

# Using DAFNI for a large-scale flood simulation demonstrator project

Robin Wardle, Research Software Engineer

National Innovation Centre for Data, Newcastle University

Systems Research Challenges Workshop, 25<sup>th</sup> - 27<sup>th</sup> April 2023

# PYRAMID Demonstrator Platform

PYRAMID: Platform for dYnamic, hyper-resolution, near-real time flood Risk AssessMent  
Integrating repurposed and novel Data sources

Collaborators

- Newcastle University Civil Engineering, Water Research Group
- Newcastle University Urban Observatory
- Loughborough University Civil Engineering
- Newcastle Innovation Centre for Data (NICD)
- External stakeholders: Newcastle City Council, National Rail, Highways Agency, Residents and Community groups

# PYRAMID - Problem

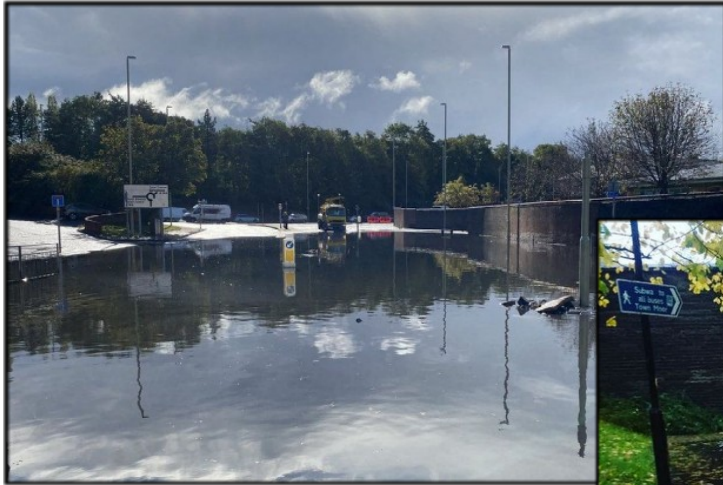
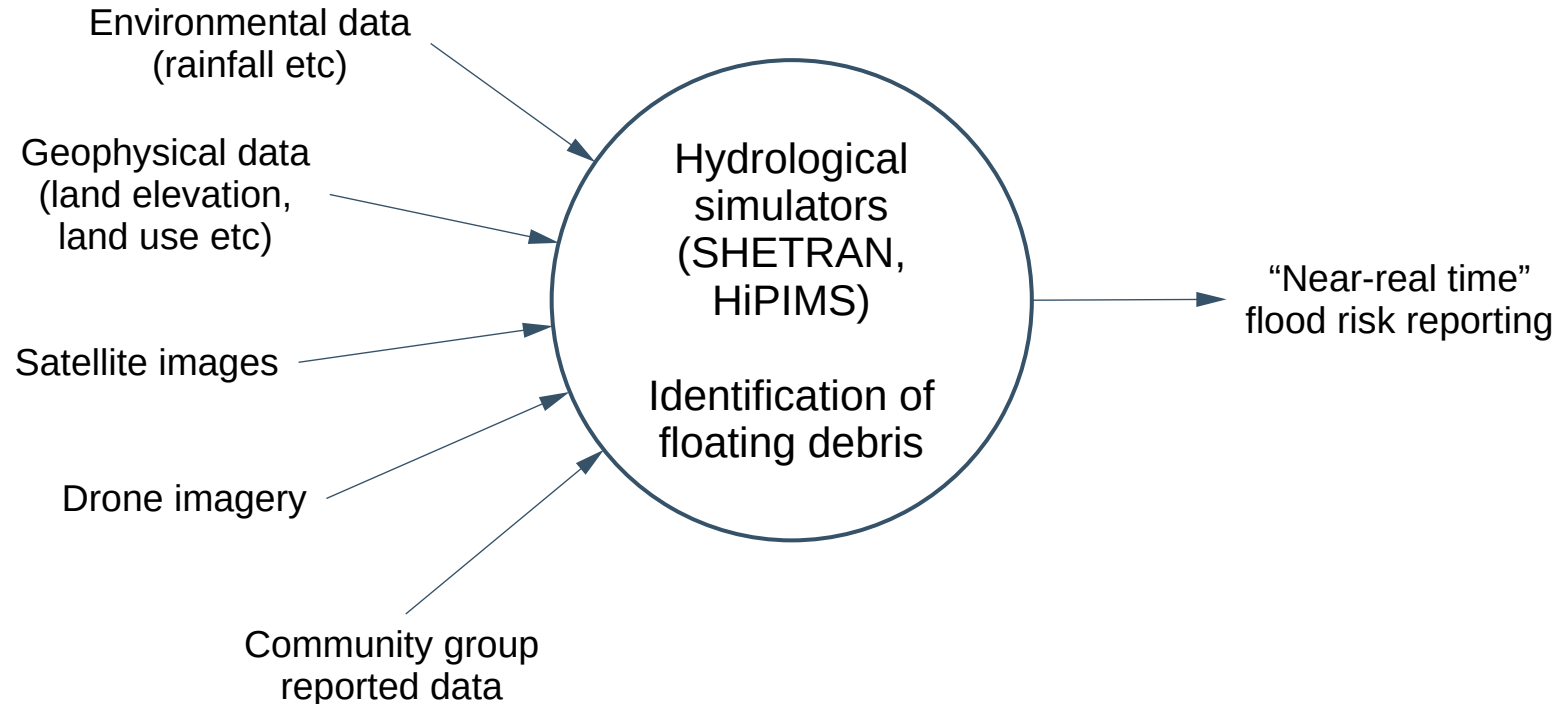


Photo sources: BBC News (Steven Lomas), ChronicleLive, Railfreight, Urban Foresight

# PYRAMID - Project Structure



## DAFNI

Data & Analytics Facility for National Infrastructure - <http://www.dafni.ac.uk>

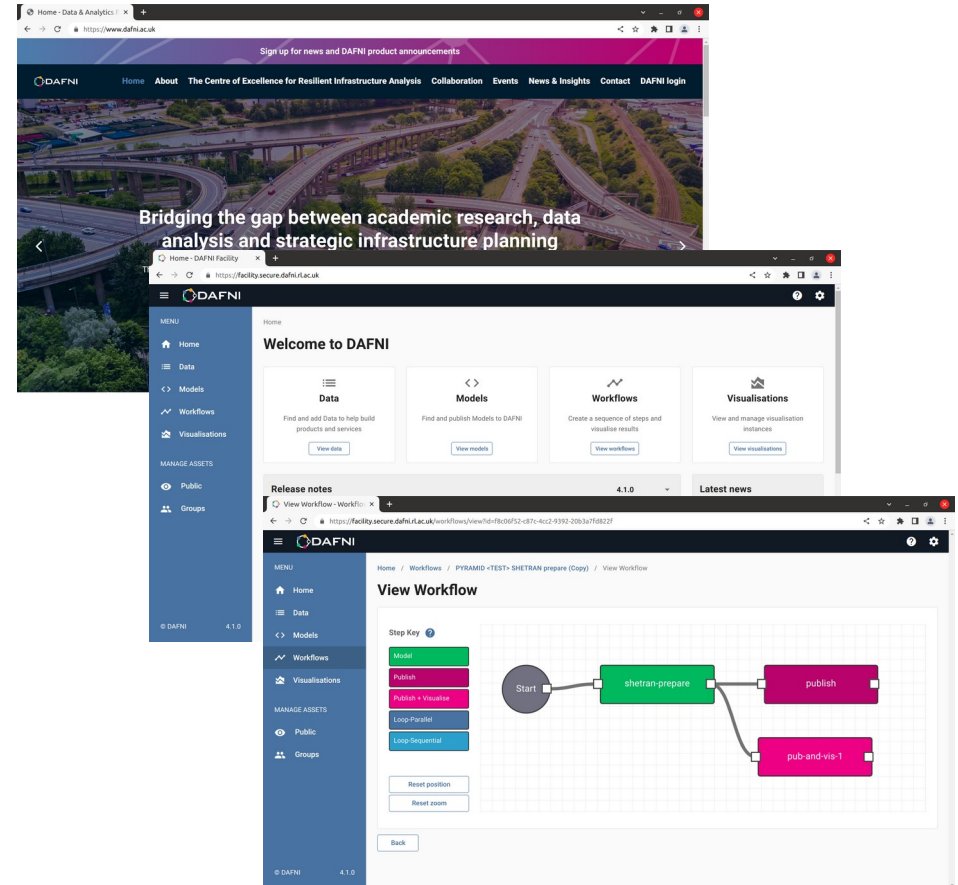
STFC Rutherford Appleton Laboratory site

Kubernetes and MinIO based platform for data and model hosting, workflow orchestration, and visualisation

792 cores, 10 GPU nodes, 2PB storage

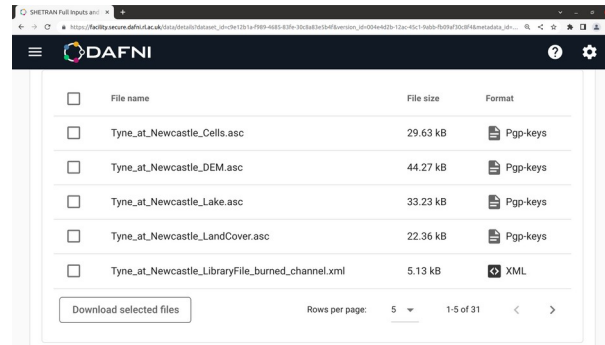
Interaction through web, API or CLI

Currently free for academic infrastructure projects



## DAFNI elements

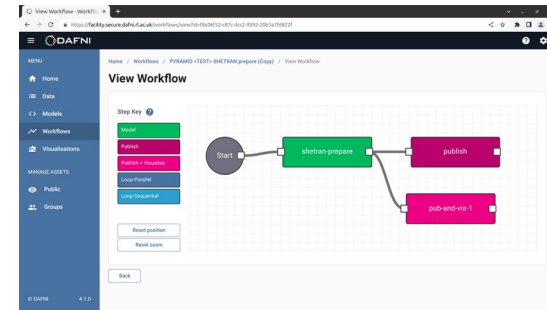
*Datasets*



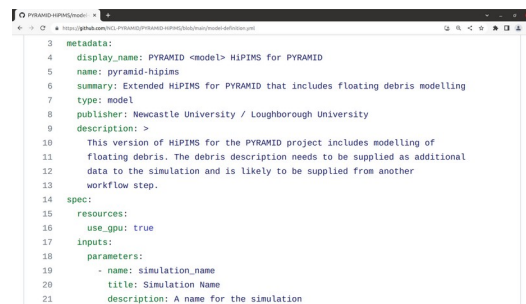
A screenshot of the DAFNI web interface showing a list of datasets. The table has columns for File name, File size, and Format. There are six rows of data, each with a checkbox in the first column. A 'Download selected files' button is at the bottom left, and 'Rows per page: 5' and '1-5 of 31' are at the bottom right.

<input type="checkbox"/>	File name	File size	Format
<input type="checkbox"/>	Tyne_at_Newcastle_Cells.asc	29.63 kB	Pgp-keys
<input type="checkbox"/>	Tyne_at_Newcastle_DEM.asc	44.27 kB	Pgp-keys
<input type="checkbox"/>	Tyne_at_Newcastle_Lake.asc	33.23 kB	Pgp-keys
<input type="checkbox"/>	Tyne_at_Newcastle_LandCover.asc	22.36 kB	Pgp-keys
<input type="checkbox"/>	Tyne_at_Newcastle_LibraryFile_burned_channel.xml	5.13 kB	XML

*Workflows*



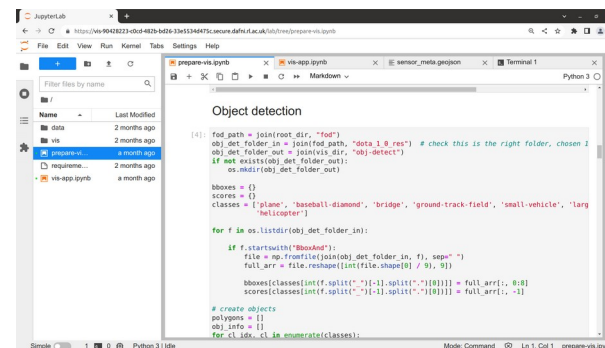
*Models*



A screenshot of the DAFNI 'Models' page showing the metadata for a model named 'pyramid-hipms'. The metadata includes display name, name, summary, publisher, description, spec, resources, inputs, and parameters.

```
3 metadata:
4 display_name: PYRAMID <model> HIPMS for PYRAMID
5 name: pyramid-hipms
6 summary: Extended HIPMS for PYRAMID that includes floating debris modelling
7 type: model
8 publisher: Newcastle University / Loughborough University
9 description: >
10 This version of HIPMS for the PYRAMID project includes modelling of
11 floating debris. The debris description needs to be supplied as additional
12 data to the simulation and is likely to be supplied from another
13 workflow step.
14 spec:
15 resources:
16 use_gps: true
17 inputs:
18 parameters:
19   - name: simulation_name
20     title: Simulation Name
21     description: A name for the simulation
```

*Visualisations*



A screenshot of a JupyterLab notebook showing Python code for object detection. The code uses OpenCV and NumPy to process an image and detect objects.

```
Object detection

[4]: fod_path = join(front_dir, "fod")
obj_det_folder_in = join(fod_path, "dota_1_8_res") # check this is the right folder, chosen I
obj_det_folder_out = join(vis_dir, "obj-detect")
if not exists(obj_det_folder_out):
    os.mkdir(obj_det_folder_out)

bboxes = {}
scores = {}
classes = ['plane', 'baseball-diamond', 'bridge', 'ground-track-field', 'small-vehicle', 'large-helicopter']

for f in os.listdir(obj_det_folder_in):
    if f.startswith("BboxAnd"):
        file = os.path.join(obj_det_folder_in, f, seqn")
        full_arr = file.reshape([int(file.shape[0]) / 9], 9))

        bboxes[classes[int(f.split("-")[1].split(".")[0])] = full_arr[:, 0:8]
        scores[classes[int(f.split("-")[1].split(".")[0])] = full_arr[:, -1]

# create objects
polygons = {}
obj_info = {}
for cls_idx, cls in enumerate(classes):
```

## PYRAMID Models (Docker images)

### Simulators / analysers

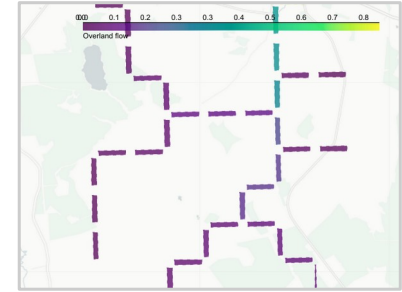
- **SHETRAN** catchment area hydrological simulator
- **HiPIMS** high-resolution water flow simulator (with floating objects)
- Floating Object Detector (**FOD**) satellite imagery processor

### Converters

- SHETRAN-to-HiPIMS
- FOD-to-HiPIMS
- Data routers and amalgamators
- Raw environmental data conversion

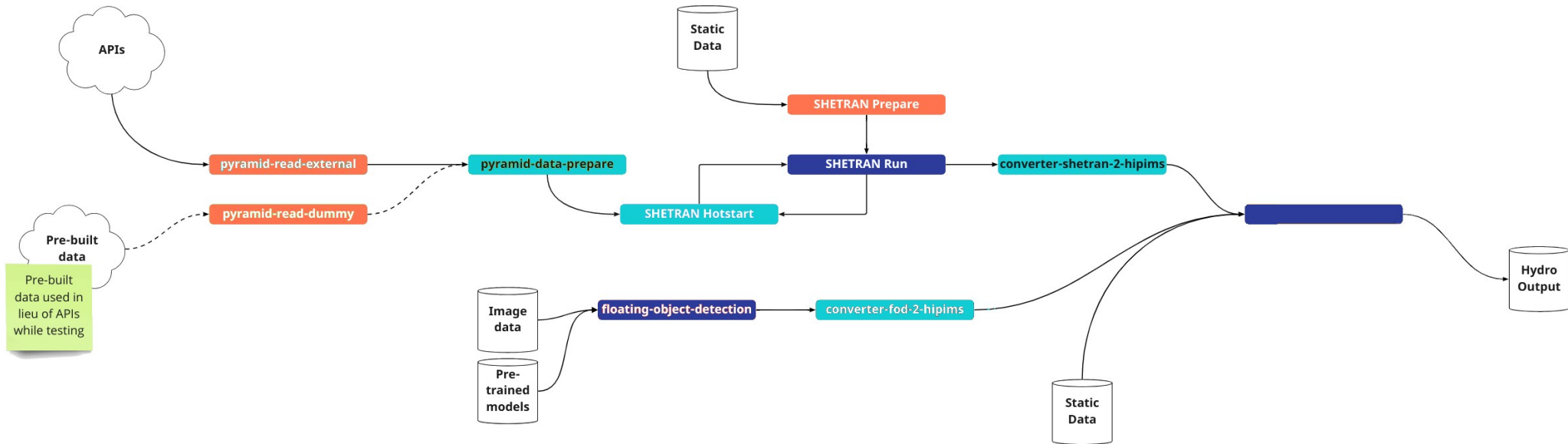
### Data wrangling

- SHETRAN environmental data preparation
- Weather sensor data API polling



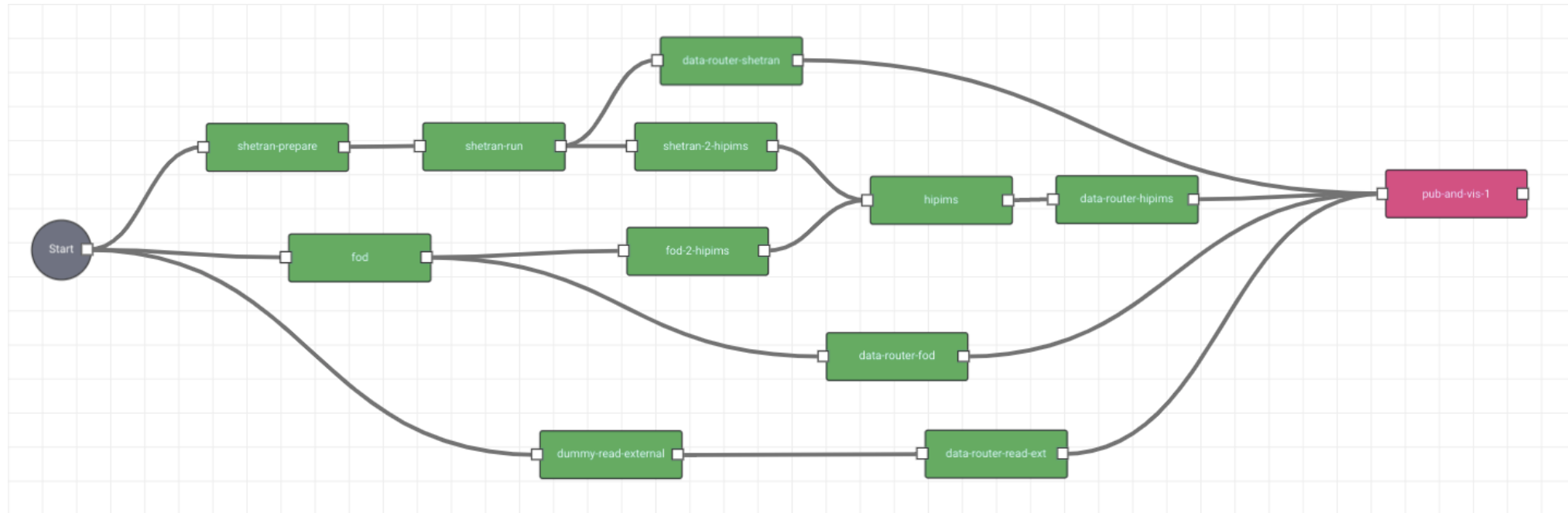
Openmap / Met Office

# Logical Architecture

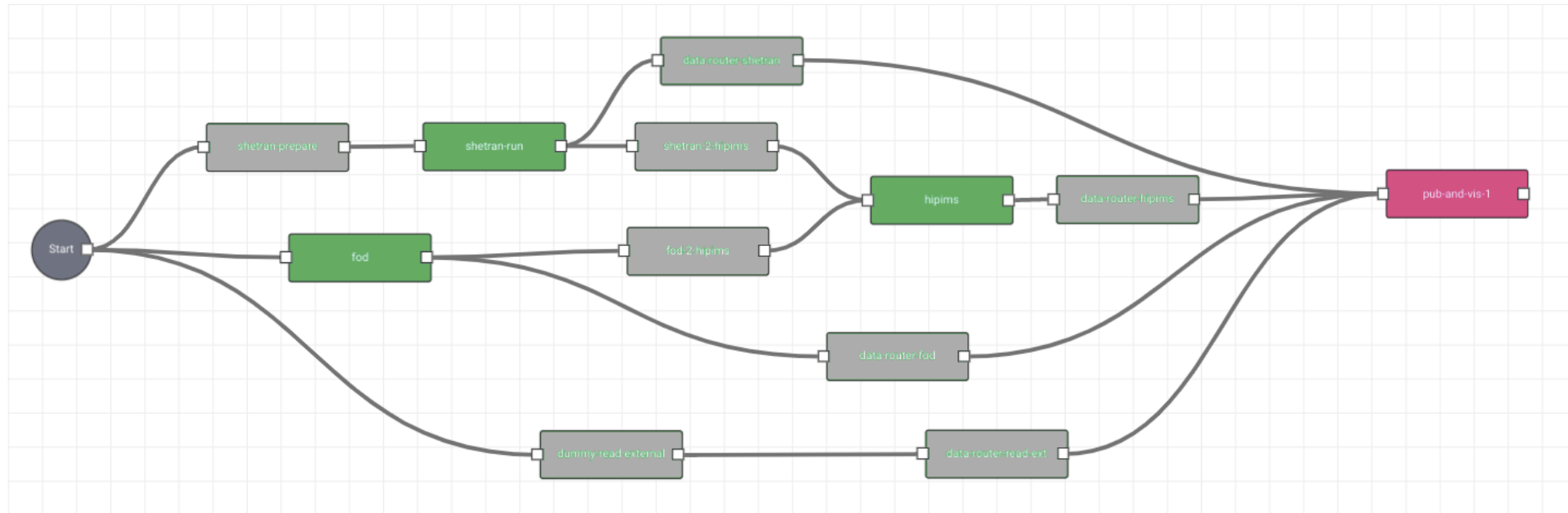




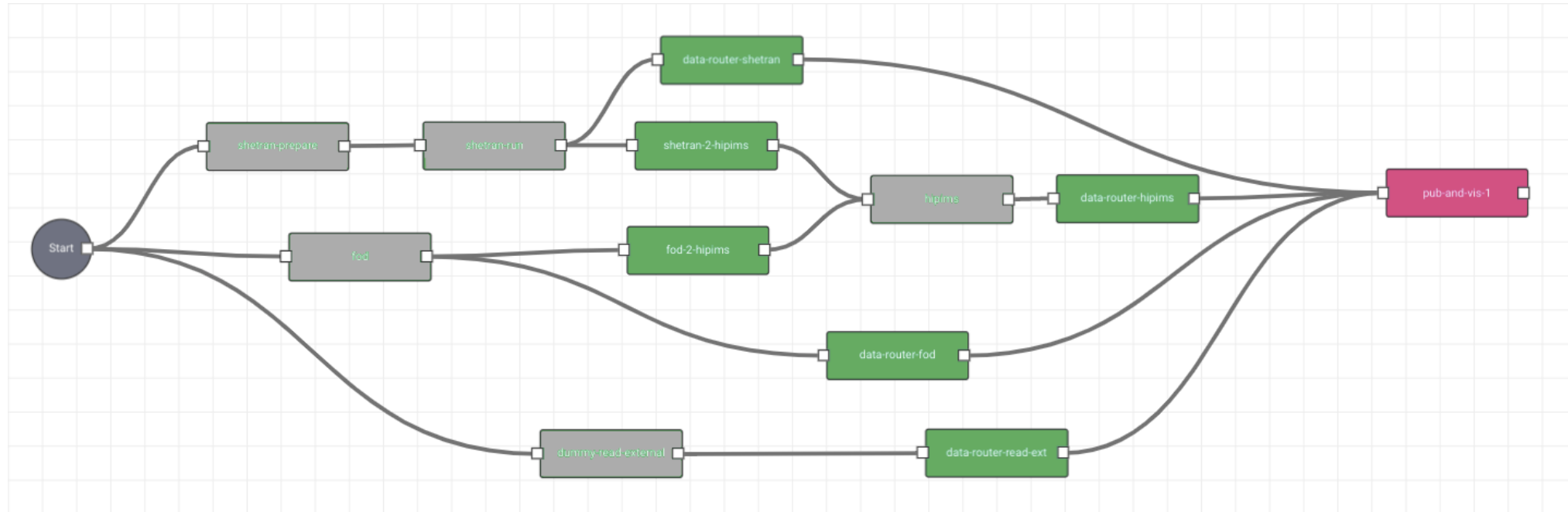
# PYRAMID DAFNI workflow - overall



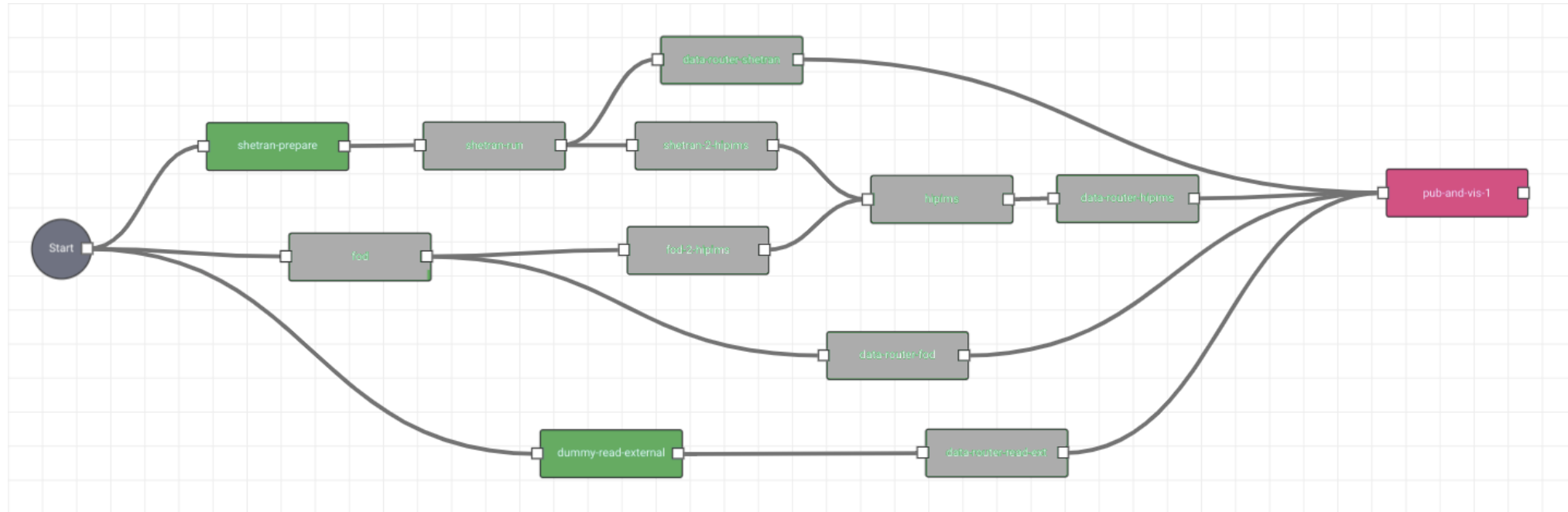
# PYRAMID DAFNI workflow - simulators



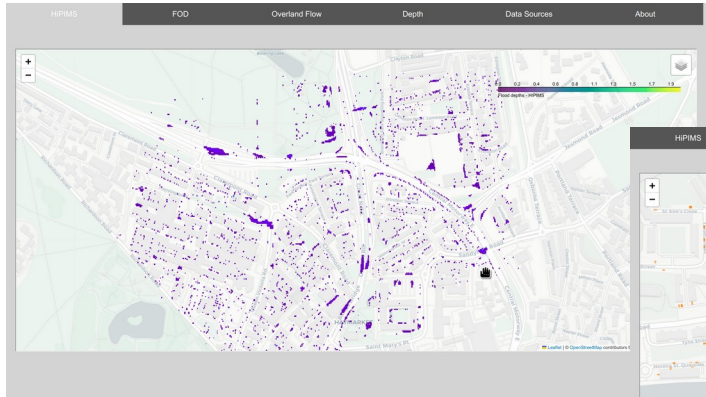
# PYRAMID DAFNI workflow - converters



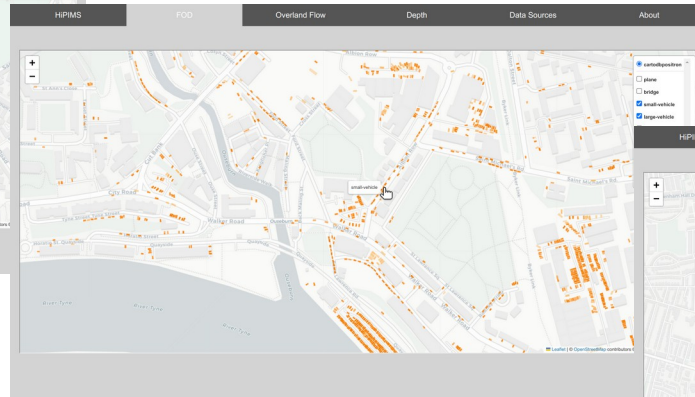
# PYRAMID DAFNI workflow - data wrangling



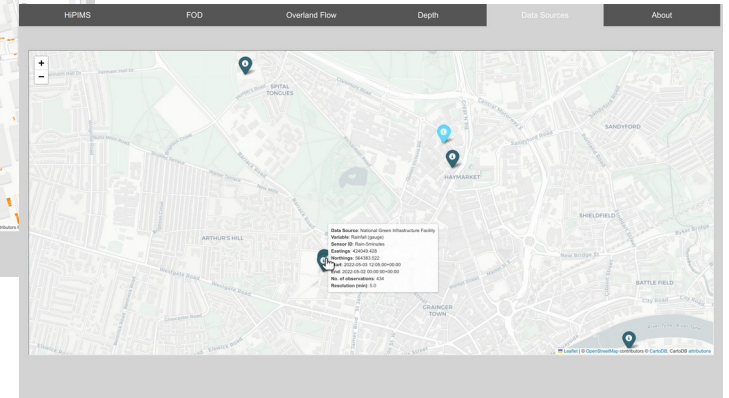
# Visualisation (Azure Web App)



Water Depth



Floating Objects  
(Vehicles)



Sensor Locations

# Complications

- HiPIMS and FOD models use GPUs (CUDA)
  - Docker image creation is very complicated for these models
  - Images are 8GB and 14GB respectively and have caused some difficulties
- Deployment using GitHub actions is a must
- Docker image creation is confusing to non-specialists (in roadmap)
- Management of researcher code (team of 4-5 RAs) is difficult
- On-platform model testing is difficult – can only be done within a workflow

# Complications

- Current DAFNI limitations
  - By default only 1 GPU available (workaround possible)
  - No loops, iteration or results caching (in roadmap?)
  - No job triggering (in roadmap)
  - Visualisation is limited to Jupyter Notebooks
  - Weak error reporting
  - Internal model-to-model data transfer is network bandwidth limited

## Future Challenges

- Streaming satellite imagery
  - large amounts of data
  - timeliness?
- Weather data reliability
- Integrating community data
  - Varying types (video, notebooks, Twitter posts, etc)
  - Not readily accessible
  - Not easily verifiable
- Workflow management
  - Job triggering
  - Loops and iteration
  - Run time
- Visualisation
- Stakeholder involvement
  - What forms of visualisation?
  - Decision-making tools
- Links to other projects:
  - OpenCLIM & CReDo



# Thanks

## DAFNI

<https://www.dafni.ac.uk>

## PYRAMID

<https://gtr.ukri.org/projects?ref=NE/V00378X/1>

## SHETRAN

<https://research.ncl.ac.uk/shetran/>

## HiPIMS

<https://www.hemlab.org/models/>

## Contact

Robin Wardle, Research Software Engineer, NICD <[robin.wardle@newcastle.ac.uk](mailto:robin.wardle@newcastle.ac.uk)>

Elizabeth Lewis, Deputy Director of Research & Innovation, School of Engineering <[elizabeth.lewis2@ncl.ac.uk](mailto:elizabeth.lewis2@ncl.ac.uk)>